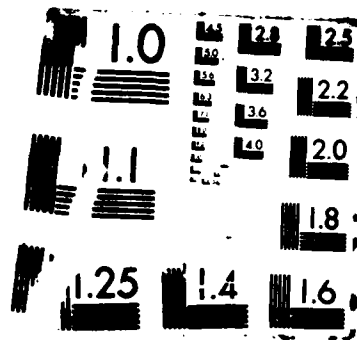


AD-A182 011 INTEGRATED INFORMATION SUPPORT SYSTEM (IIS) VOLUME 5 1/1
COMMON DATA MODEL 8... (U) GENERAL ELECTRIC CO
SCHENECTADY NY PRODUCTION RESOURCES CONSU..
UNCLASSIFIED J L ALTHOFF ET AL. 01 NOV 85 PS-620141259 F/G 12/5 NL

END

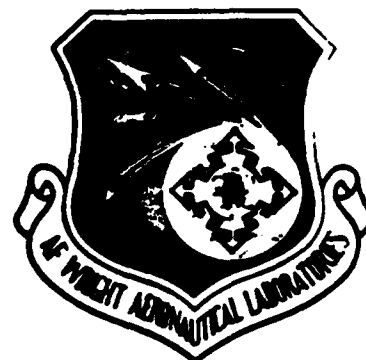
DATE
FILMED
62



AD-A182 011

DTIC FILE COPY

**AFWAL-TR-86-4006
Volume V
Part 23**



**INTEGRATED INFORMATION
SUPPORT SYSTEM (IISS)
Volume V - Common Data Model Subsystem
Part 23 - NDML Precompiler Build Source Code
Product Specification**

**General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345**

**Final Report for Period 22 September 1980 - 31 July 1985
November 1985**

Approved for public release; distribution is unlimited.

**MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533**

**DTIC
ELECTE
JUL 02 1987
S E D**

87 6 30 117

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

A182 011

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.	
7a. DECLASSIFICATION/DOWNGRADING SCHEDULE			
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFVAL-TR-86-4006 Vol V, Part 23	
6a. NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting	6b. OFFICE SYMBOL (If applicable) AFVAL/MLTC	7b. NAME OF MONITORING ORGANIZATION AFVAL/MLTC	
6c. ADDRESS (City, State and ZIP Code) 1 River Road Schenectady, NY 12345		7c. ADDRESS (City, State and ZIP Code) WPAFB, OH 45433-8533	
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Materials Laboratory Air Force Systems Command, USAF	8b. OFFICE SYMBOL (If applicable) AFVAL/MLTC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-80-C-5185	
8c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433		10. SOURCE OF FUNDING NOS.	
		PROGRAM ELEMENT NO. Y8011F	PROJECT NO. 7500
		TASK NO. 62	WORK UNIT NO. 01
11. TITLE (Include Security Classification) (See Reverse)			
12. PERSONAL AUTHOR(S) Althoff, J. L. and Apicella, M. L.			
13a. TYPE OF REPORT Final Technical Report	13b. TIME COVERED 22 Sept 1980 - 31 July 1985	14. DATE OF REPORT (Yr., Mo., Day) 1985 November	15. PAGE COUNT 40
16. SUPPLEMENTARY NOTATION ICAM Project Priority 6201 The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB GR	
1308	0905		
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>This document is the product specification establishing the design implementation of the IISS Configuration Item PRE11 which will recombine the original user input program along with generated modifications into a single program file.</p> <p style="text-align: right;">→ See p1-1</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson		22b. TELEPHONE NUMBER (Include Area Code) 813-255-8876	22c. OFFICE SYMBOL AFVAL/MLTC

DO FORM 1473, 83 APR

EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

11. Title

Integrated Information Support System (IISS)
Vol V - Common Data Model Subsystem
Part 23 - NDML Precompiler Build Source Code
Product Specification

A S D 86 1457
17 Jul 1986

Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input checked="checked" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



PS 620141259
1 November 1985

PREFACE

This product specification covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

Subcontractors

Role

Boeing Military Aircraft
Company (BMAC)

Reviewer.

D. Appleton Company
(DACOM)

Responsible for IDEF support,
state-of-the-art literature
search.

General Dynamics/
Ft. Worth

Responsible for factory view
function and information
models.

PS 620141259
1 November 1985

<u>Subcontractors</u>	<u>Role</u>
Illinois Institute of Technology	Responsible for factory view function research (IITRI) and information models of small and medium-size business.
North American Rockwell	Reviewer.
Northrop Corporation	Responsible for factory view function and information models.
Pritsker and Associates	Responsible for IDEF2 support.
SofTech	Responsible for IDEFO support.

TASKS 4.3 - 4.9 (TEST BED)

<u>Subcontractors</u>	<u>Role</u>
Boeing Military Aircraft Company (BMAC)	Responsible for consultation on applications of the technology and on IBM computer technology.
Computer Technology Associates (CTA)	Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager.
Control Data Corporation (CDC)	Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM).
D. Appleton Company (DACOM)	Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems.

PS 620141259
1 November 1985

<u>Subcontractors</u>	<u>Role</u>
Digital Equipment Corporation (DEC)	Consulting and support of the performance testing and on DEC software and computer systems operation.
McDonnell Douglas Automation Company (McAuto)	Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.
On-Line Software International (OSI)	Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.
Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge)	Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.
SofTech, Inc.	Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.
Software Performance Engineering (SPE)	Responsible for directing the work on performance evaluation and analysis.
Structural Dynamics Research Corporation (SDRC)	Responsible for the User Interface and Virtual Terminal Interface Subsystems.

Other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC).

PS 620141259
1 November 1985

<u>Contractors</u>	<u>ICAM Project</u>	<u>Contributing Activities</u>
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP).
D. Appleton Company (DACOM)	1502	IISS enhancements to Integration Methodology.
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements.
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1703	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI).
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

SECTION 1.0	SCOPE	1-1
1.1	Identification	1-1
1.2	Functional Summary	1-1
SECTION 2.0	DOCUMENTS	2-1
2.1	Reference Documents	2-1
2.2	Terms and Abbreviations	2-1
SECTION 3.0	REQUIREMENTS	3-1
3.1	Structural Description	3-1
3.2	Functional Flow	3-1
3.3	Interfaces	3-1
3.3.1	Inputs/Outputs	3-2
3.4	Program Interrupts	3-2
3.5	Timing and Sequencing Description	3-2
3.6	Special Control Features	3-2
3.7	Storage Allocation	3-2
3.7.1	Database Definition	3-2
3.7.1.1	File Description	3-2
3.7.1.2	Table Description	3-3
3.7.1.3	Item Description	3-3
3.8	Object Code Creation	3-3
3.9	Adaptation Data	3-3
3.10	Detail Design Description	3-3
3.10.1	Main Program List	3-3
3.10.2	Module List	3-5
3.10.3	External Routines List	3-7
3.10.4	Include File List	3-9
3.10.5	Where Include File Used List	3-11
3.10.6	Where External Routine Used List ...	3-13
3.10.7	Main Program Parts List	3-16
3.10.8	Module Documentation	3-18
3.10.9	Include File Description	3-22
3.10.10	Hierarchy Chart	3-23
3.11	Program Listings Comments	3-27
SECTION 4.0	QUALITY ASSURANCE PROVISIONS	4-1
4.1	Introduction and Definitions	4-1
4.2	Computer Programming and Test Evaluation	4-1

SECTION 1

SCOPE

1.1 Identification

This specification establishes the design of Function PRE11, "Build Source Code", one of the major functions of the Configuration Item "Precompiler" to be built and formally accepted by the ICAM Program Office. This CI constitutes one of the subsystems of the Common Data Model Processor (CDMP).

1.2 Functional Summary

The purpose of this Computer Program Configuration Item (CPCI) is to combine previously constructed parcels into a modified application process capable of servicing NDML requests. The following functions will be performed by this CPCI:

1. Write contents of parcel 2 onto the first parcel.
2. Write contents of parcel 3 onto the first parcel.
3. Write contents of parcel 4 onto the first parcel.
4. Delete parcels 1, 2 and 3.

*Keywords: ICAM Integrated
Computer Aided Manufacturing*

SECTION 2

DOCUMENTS

2.1 Reference Documents

1. ICAM Documentation Standards: IDS15012000A, 28 December 1981.
2. D. Appleton Co., CDM Administrators Manual; UM620141000, March 1984.
3. D. Appleton Co., CDM1-IDEF, Model of the Common Data Model; CCS620141000, 15 May 1985.
4. D. Appleton Co., Computer Program Development Specification (DS) for ICAM Integrated Support System (IISS) Configuration Item: NDML Precompiler; DS620141200, October 1984.
5. D. Appleton Co., Embedded NDML Programmer's Reference Manual; PRM620141200, March 1985
6. Softech, Inc., NTM Programmer's Guide; UM620140001, July, 1984.
7. Control Data Corp., Computer Program Development Specification (DS) for ICAM Integrated Support System (IISS) Configuration Item: NDDL Command Processor; DS620141100, June 1985.

2.2 Terms and Abbreviations

Attribute Use Class: (AUC)

Conceptual Schema: (CS)

Common Data Model Processor: (CDMP)

Common Data Model: (CDM) Describes common data application process formats, form definitions, etc., of the IISS and includes conceptual schema, external, internal schemas, and schema transformation operators.

Data Field: (DF) An element of data in the external schema. It is by this name that an NDML programmer references

data.

Database Management System: (DBMS)

Distributed Request Supervisor: (DRS) This IISS CDM subsystem configuration item controls the execution of distributed NDML queries and non distributed updates.

Domain: A logical definition of legal attribute class values.

Domain Constraint: Predicate that applies to a single domain.

External Schema: (ES)

Forms: Structured views which may be imposed on windows or other forms. A form is composed of fields where each field is a form, item, or window.

Forms Processor: (FP) A set of callable execution time routines available to an application program for form processing.

Internal Schema: (IS)

Integrated Information Support System: (IISS) A test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous databases supported by heterogeneous computers interconnected via a local Area Network.

Mapping: The correspondence of independent objects in two schemas: ES to CS or CS to IS.

Network Transaction Manager: (NTM) Performs the coordination, communication and housekeeping functions required to integrate the application processes and system services resident on the various hosts into a cohesive system.

Neutral Data Manipulation Language: (NDML) A language developed by the IISS project to provide uniform access to common data, regardless of database manager or distribution criteria. It provides distributed retrieved and single node updates.

PS 620141259
1 November 1985

ORACLE: Relational DBMS based on the SQL (Structured Query Language, a product of ORACLE Corp, Menlo Park, CA). The CDM is an ORACLE database.

Parcel: A sequential file containing section source code of the input application program.

Request Processor: (RP) A COBOL program that will satisfy a retrieval or update NDML subtransaction against a particular Database Management System.

User Interface: (UI) Controls the user's terminal and interfaces with the rest of the system.

Virtual Terminal Interface: (VTI) Performs the interfacing between different terminals and the UI. This is done by defining a specific set of terminal features and protocols which must be supported by UI software which constitutes the Virtual Terminal Definition. Specific terminals are then mapped against the Virtual Terminal software by specific software modules written for each type of real terminal supported.

SECTION 3

REQUIREMENTS

3.1 Structural Description

A graphic portrayal of this CPCI is included in Section 3.10. This chart shows the hierarchical relationship of each module making up this CPCI. As can be seen, the lower level routines Open, Read and Close the appropriate files. The Delete file interface is used to remove the parcels that were copied into the first parcel.

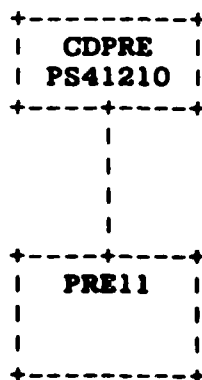
3.2 Functional Flow

This CPCI implements the logic defined in the Development Specification for this CPCI. Details of inputs/outputs and relationships between modules are to be found in Section 3.10.

This CPCI has been designated to operate in a batch or interactive mode. It must operate in the system environment established for IISS; that is, use of the Network Transaction Manager. It must use the ORACLE Database Management System installed on a DEC VAX computer.

3.3 Interfaces

The following diagram depicts interface of PRE11 with other CPCI's in the system.



3.3.1 Inputs/Outputs

The following table depicts the inputs and outputs of this CPCI. A detail description for each item can be found in the DS for this CPCI.

Function: PRE11

INPUT	OUTPUT
Parcel 1 File Name	Module Status
Parcel 2 File Name	
Parcel 3 File Name	
Parcel 4 File Name	
Source Language	
Current Host	

3.4 Program Interrupts

Not applicable to this CPCI.

3.5 Timing and Sequencing Description

This module is called under the control of CDPRE, the precompiler control module. PRE11 is called once per successful precompilation of a single user module.

3.6 Special Control Features

Not applicable to this CPCI.

3.7 Storage Allocation

3.7.1 Database Definition

The database used by this CPCI is the Common Data Model (CDM) database. This model is defined by the CDM1, the IDEF-1 model of the CDM, Reference Document Number 3.

3.7.1.1 File Description

No permanent files have been defined for this CPCI. It may

use temporary scratch files for such things as generated program source code or temporary query results.

3.7.1.2 Table Description

All tables used by this CPCI have been defined by the Development Specification for this CPCI.

3.8 Object Code Creation

The object code for this CPCI will be created by the system integration test team by using defined IISS Software Configuration Management Procedures. This CPCI will use the "C" language compilers.

3.9 Adaptation Data

This CPCI has been coded using ANSI COBOL, FORTRAN and a "standard" subset of the "C" language. The intent was to provide a transportable system. Any system environment supporting this language, a virtual memory management scheme, the COMM and NTM subsystems of IISS and the ORACLE Database Management System should be able to support this CPCI. Every possible attempt has been made to localize and identify any machine or environment dependent modules through the original design of the IISS and application of Configuration Management Procedures.

3.10 Detail Design Description

The following sections have been computer generated for this CPCI.

3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it is truly a main program entry point. If not, then it is merely called by other programs not being documented here.

PS 620141259
1 November 1985

BUILD SOURCE CODE Main Program List

Module Name -----	Purpose -----
CDP12	CDP12 APPENDS PARCL2, PARCL3, PARCL4 TO THE PARCL1

PS 620141259
1 November 1985

3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

PS 620141259
1 November 1985

BUILD SOURCE CODE Module List

Module Name -----	Purpose -----
CDP12	CDP12 APPENDS PARCL2, PARCL3, PARCL4 TO THE PARCL1

PS 620141259
1 November 1985

3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. The specification in which any module is documented may be found in the Module Documentation Index (Document Number CM 620100001). See section 3.10.6 for a list of the modules that call each of these external routines.

PS 620141259
1 November 1985

BUILD SOURCE CODE External Routines List

Module Name -----	First User -----
CLSFILE	CDP12
DELFIL	CDP12
ERRPRO	CDP12
FOPEN	CDP12
REDLINE	CDP12
SPRINTF	CDP12
STRCPY	CDP12
STRNCMP	CDP12
WRTLINE	CDP12

3.10.4 Include File List

The following is a list of all include files called in by modules being documented here. Each include file has a unique name regardless of the language being used. The purpose of each include file is listed as well. A more complete description of each include file is given in section 3.10.9. The purpose listed is the one that is in the source code of the include file.

A purpose of "**** PURPOSE NOT FOUND BY STRIPPER ****" indicates that a purpose statement was not written into the include file itself. The most common reason for this is that the include file comes from system libraries that were not developed by the project, such as 'C' libraries that are provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all the modules which call in each of these include files.

PS 620141259
1 November 1985

BUILD SOURCE CODE Include File List

File Name

Purpose

STDIO

**** PURPOSE NOT FOUND BY STRIPPER ****

PS 620141259
1 November 1985

3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

PS 620141259
1 November 1985

BUILD SOURCE CODE Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
STDIO	CDP12	CDP12 APPENDS PARCL2, PARCL3, PARCL4 TO THE PARCL1

PS 620141259
1 November 1985

3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.

PS 620141259
1 November 1985

BUILD SOURCE CODE Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
CLSFILE	CDP12	CDP12 APPENDS PARCL2, PARCL3,PARCL4 TO THE PARCL1
DELFIL	CDP12	CDP12 APPENDS PARCL2, PARCL3,PARCL4 TO THE PARCL1
ERRPRO	CDP12	CDP12 APPENDS PARCL2, PARCL3,PARCL4 TO THE PARCL1
FOPEN	CDP12	CDP12 APPENDS PARCL2, PARCL3,PARCL4 TO THE PARCL1
REDLINE	CDP12	CDP12 APPENDS PARCL2, PARCL3,PARCL4 TO THE PARCL1
SPRINTF	CDP12	CDP12 APPENDS PARCL2, PARCL3,PARCL4 TO THE PARCL1
STRCPY	CDP12	CDP12 APPENDS PARCL2, PARCL3,PARCL4 TO THE PARCL1

PS 620141259
1 November 1985

BUILD SOURCE CODE Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
STRNCMP	CDP12	CDP12 APPENDS PARCL2, PARCL3, PARCL4 TO THE PARCL1
WRTLINE	CDP12	CDP12 APPENDS PARCL2, PARCL3, PARCL4 TO THE PARCL1

3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more than once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external routine". The Purpose of the Main Program module is listed as well.

PS 620141259
1 November 1985

BUILD SOURCE CODE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
CDP12	Purpose-->	CDP12 APPENDS PARCL2, PARCL3, PARCL4 TO THE PARCL1
	CLSFILE	External routine
	DELFIL	External routine
	ERRPRO	External routine
	FOPEN	External routine
	REDLINE	External routine
	SPRINTF	External routine
	STRCPY	External routine
	STRNCMP	External routine
	WRTLINE	External routine

3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME:	Name of program Module.
PURPOSE:	Purpose of Module as detailed in the source code.
LANGUAGE:	Programming language source code is written in. The choices are: VAX-11 FORTRAN C (I/S-1 Workbench 'C') VAX-11 COBOL
MODULE TYPE:	Whether a Program, Subroutine, or Function.
SOURCE FILE:	Name of Source File from file specification.
SOURCE FILE TYPE:	Source File Extension from file specification.
HOST:	Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
SUBSYSTEM:	IISS sub-system this file resides in.
SUBDIRECTORY:	Sub-directory of that subsystem in which this file resides.
DOCUMENTATION GROUP:	Name of documentation group of which this source file is a member.
DESCRIPTION:	A description of the module as obtained

PS 620141259
1 November 1985

from the source code.

ARGUMENTS: The arguments with which this routine is called if it is a Subroutine or a Function.

INCLUDE FILES: A list of all the files that are included into this module as well as their purposes.

ROUTINES CALLED: Subroutines or Functions, either documented or external, called by this module, if any.

CALLED DIRECTLY BY: The documented routines which call this module, if any.

USED IN MAIN PROGRAM(S): The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.

PS 620141259
1 November 1985

BUILD SOURCE CODE Module Documentation

NAME: CDP12
PURPOSE: CDP12 APPENDS PARCL2, PARCL3, PARCL4 TO
THE PARCL1
LANGUAGE: C
MODULE TYPE: FUNCTION
FUNCTION TYPE: INT ()
SOURCE FILE: CDP12
SOURCE FILE TYPE: .C
HOST:
SUBSYSTEM: CDM
SUBDIRECTORY:
DOCUMENTATION GROUP: PS41259

DESCRIPTION:

SYNOPSIS

C

CDP12(PARCL1, PARCL2, PARCL3, PARCL4, LANGUAGE, FILE_HOST,

COBOL -- CALL "CDP12" RET-STATUS) ;
USING

PARCL1,
PARCL2,
PARCL3,
PARCL4,

LANGUAGE,

FILE-HOST,

RET-STATUS.

FORTRAN -- CALL
CDP12(PARCL1, PARCL2, PARCL3, PARCL4, LANGUAGE,
FILEHOST, RET-STATUS)

INPUT:
CHAR

*PARCL1 ;

PS 620141259
1 November 1985

CHAR	*PARCL2 ;
CHAR	*PARCL3 ;
CHAR	*PARCL4 ;
CHAR	*LANGUAGE ;
CHAR	*FILE_HOST ;

OUTPUT:
CHAR *RET-STATUS ;

DESCRIPTION

CDP12 -- CDP12 APPENDS PARCL2, PARCL3, PARCL4 TO THE
PARCL1
SO THE PARCL1 WILL BE A COMPLETE PROGRAM.

ARGUMENTS:

PARCL1 -	CHAR *
PARCL2 -	CHAR *
PARCL3 -	CHAR *
PARCL4 -	CHAR *
LANGUAGE -	CHAR *
FILE HOST -	CHAR *
STATUS -	CHAR *

INCLUDE FILES:

STDIO - **** PURPOSE NOT FOUND BY STRIPPER ****

ROUTINES CALLED:

STRCPY
FOPEN
SPRINTF
ERRPRO
REDLINE
WRTLINE
CLSFILE
DELFIL
STRNCMP

PS 620141259
1 November 1985

3.10.9 Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code. The language it is written in is also given.

3.10.10 Hierarchy Chart

The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the first page are considered Main Programs as described in section 3.10.1 above.

There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.

PS 620141259
1 November 1985

1

+-----+
|CDP12|
+-----+

|

+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
STRCPY	FOPEN	SPRINTF	ERRPRO	REDLINE	(CONT)
+-----+	+-----+	+-----+	+-----+	+-----+	+-----2+

PS 620141259
1 November 1985

2

```

      +-----+
      | CDP12 |
      +-----+
        |
+-----+-----+-----+-----+-----+
|         |         |         |         |         |
+-----+ +-----+ +-----+ +-----+ +-----+
| (CONT) | | WRTLINE | | CLSFILE | | DELFIL | | STRNCMP |
+-----+ +-----+ +-----+ +-----+ +-----+

```

PS 620141259
1 November 1985

CDP12.....1
CLSFILE
DELFIL
ERRPRO
FOPEN
REDLINE
SPRINTF
STRCPY
STRNCMP
WRTLINE

PS 620141259
1 November 1985

3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

SECTION 4
QUALITY ASSURANCE PROVISIONS

4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."

END

DATE
FILMED

5-87